

SOLUÇÃO HEURÍSTICA PARA O
PROBLEMA DO CAIXEIRO VIAJANTE

UNIVERSIDADE FEDERAL DE SANTA CATARINA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DE PRODUÇÃO

SOLUÇÃO HEURÍSTICA PARA O
PROBLEMA DO CAIXEIRO VIAJANTE

DISSERTAÇÃO SUBMETIDA À UNIVERSIDADE FEDERAL DE SANTA CATARINA
PARA A OBTENÇÃO DO GRAU DE
"MESTRE EM ENGENHARIA"

EUGÊNIO RUBENS CARDOSO BRAZ

FLORIANÓPOLIS
SANTA CATARINA - BRASIL
AGOSTO - 1980

SOLUÇÃO HEURÍSTICA PARA O
PROBLEMA DO CAIXEIRO VIAJANTE

EUGÊNIO RUBENS CARDOSO BRAZ

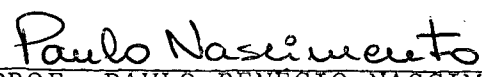
ESTA DISSERTAÇÃO FOI JULGADA PARA A OBTENÇÃO DO TÍTULO DE:
"MESTRE EM ENGENHARIA"

ESPECIALIDADE ENGENHARIA DE PRODUÇÃO, E APROVADA EM SUA FORMA
FINAL PELO PROGRAMA DE PÓS-GRADUAÇÃO.

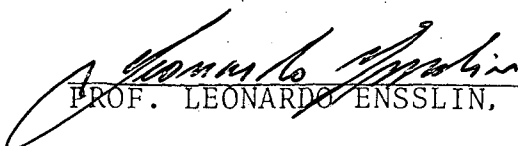


PROF. JOHN ROBERT MACKNESS, Ph.D.
COORDENADOR

APRESENTADA PERANTE A BANCA EXAMINADORA COMPOSTA PELOS PROFESSORES:



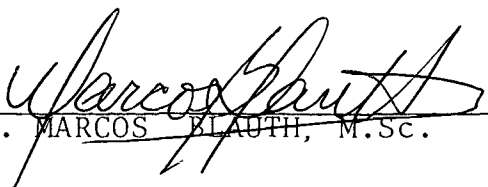
PROF. PAULO RENÉCIO NASCIMENTO, M.Sc.
PRESIDENTE



PROF. LEONARDO ENSSLIN, Ph.D.



PROF. RAUL VALENTIM DA SILVA, M.Sc.



PROF. MARCOS KLAUTH, M.Sc.



0.249.246-7

UFSC-BU

Ao meu pai,

Pedro de Alcântara Braz,
em memória.

A G R A D E C I M E N T O

Ao computador IBM/370-148 instalado na ELETROSUL, pelas noites agradáveis que me propiciou e pela participação ativa na resolução de todos os problemas.

Ao Prof. PAULO RENÉCIO NASCIMENTO, pela orientação no desenvolvimento do trabalho.

A minha filha MELISSA MEDEIROS BRAZ, por ter me ajudado, ficando junto a mim, nas fases de estudo e pesquisa de modelos.

R E S U M O

Este trabalho analisa o problema do caixeiro viajante, para o qual é desenvolvido e implementado um algoritmo de baixa complexidade tempo e baixa complexidade memória.

Inicialmente, o problema é definido e são tecidas considerações sobre a complexidade de algoritmos.

Posteriormente, é apresentada uma síntese dos métodos de resolução existentes.

A seguir é desenvolvido e apresentado o algoritmo e finalmente são relacionados alguns dos problemas estudados, juntamente com os dados de performance obtidos.

A B S T R A C T

This work deals with the traveling salesman problem, for which an algorithm, with low level complexity as far as time and memory are concerned, is developed and programmed.

First of all, the problem is defined and some considerations are stated about the complexity of algorithms.

Afterwards, a summary of the existing resolution methods is presented.

The algorithm is then developed and described, and finally, some case studies are listed together with the output performance data.

S U M Á R I O

CAPÍTULO I

Página

| | |
|-----------------------------------|---|
| 1. INTRODUÇÃO | 1 |
| 1.1. O Problema | 1 |
| 1.2. O Mundo dos Algoritmos | 2 |
| 1.3. Objetivo do Trabalho | 8 |

CAPÍTULO II

| | |
|---|----|
| 2. CONSIDERAÇÕES CONCEITUAIS | 9 |
| 2.1. Métodos de Resolução Existentes | 10 |
| 2.1.1. Melhoramento Circuito a Circuito ... | 11 |
| 2.1.2. Construção do Circuito | 12 |
| 2.1.3. Eliminação de Sub-Circuitos | 13 |

CAPÍTULO III

3. MODELO PROPOSTO

| | |
|---|----|
| 3.1. O Algoritmo | 15 |
| 3.1.1. O Ponto de Partida | 15 |
| 3.1.2. O Esquema de Geração de Soluções ... | 15 |
| 3.1.3. A Regra de Parada | 16 |

| | |
|--|----|
| 3.2. O Fluxograma | 16 |
| 3.3. Considerações sobre a Implementação | 17 |
| 3.4. Análise dos Resultados | 18 |
| 3.5. Complexidades Tempo e Memória | 19 |

CAPÍTULO IV

| | |
|------------------------------|----|
| 4. CONCLUSÃO | 21 |
| BIBLIOGRAFIA UTILIZADA | 22 |

ANEXO 1 - Listagem do Programa

ANEXO 2 - Instruções de Utilização

ANEXO 3 - Problemas de Teste

Se teus projetos

são para um ano,

- semeia o grão.

Se são para dez anos,

- planta uma árvore.

Se são para cem anos,

- instrue o povo.

Semeando o grão,

- colherás uma vez.

Plantando uma árvore,

- colherás dez vezes.

Instruindo o povo,

- colherás cem vezes.

Se deres um peixe

a um homem,

- ele se alimentará uma vez;

Se o ensinares a pescar,

- ele se alimentará toda a vida.

Kuan-Tzu, sábio chinês, séc. VII a.C.

C A P Í T U L O I

1. INTRODUÇÃO

O problema do caixeiro viajante é um problema combinatório clássico, que vem sendo estudado há décadas por pesquisadores das áreas de ciências exatas e da computação, juntamente com os problemas do circuito de Hamilton e da programação linear inteira, que estão enquadrados na mesma classe.

Apesar da ilusória idéia de inutilidade, o problema do caixeiro viajante é um excelente modelo, de grande importância econômica na solução de problemas de planejamento e programação da produção, entre outros.

1.1. O Problema

Um vendedor precisa elaborar um plano de viagem para visitar, economicamente, cada uma das n cidades onde tem clientes e retornar à cidade de onde partiu.

O termo "economicamente" pode estar relacionado ao tempo dispendido, a distância percorrida, ou mesmo ao custo total da viagem.

Em todos estes casos, deseja-se a minimização do parâmetro que for considerado.

1.2. O Mundo dos Algoritmos

O estudo de algoritmos é a parte mais emocionante da ciência da computação. Nestes últimos anos houve avanços significantes neste campo, que variaram desde o desenvolvimento de algoritmos cada vez mais rápidos, até a descoberta assustadora de certos problemas naturais, frente aos quais todos os algoritmos conhecidos mostram-se ineficientes.¹

Estes resultados aumentaram consideravelmente a motivação pelo estudo de algoritmos, e desta forma a área de projeto e análise de algoritmos tornou-se um campo de intenso interesse.

Os algoritmos podem ser avaliados por uma variedade de critérios. O mais significativo deles é a taxa do crescimento do tempo e da memória, necessária para resolver problemas cada vez maiores.

Para adotar este critério, associa-se ao problema um número inteiro, chamado de "tamanho" do problema, que representa uma medida da quantidade dos dados de entrada. Por exemplo, o "tamanho" de um problema de multiplicação de matrizes pode ser a maior dimensão de uma das matrizes a ser multiplicada. O "tamanho" de um problema de grafo pode ser o número de arcos e, no caso do caixeiro viajante, o "tamanho" do problema pode ser o número de cidades a serem visitadas.

¹

AHO, HOPCROFT & ULLMANN, Design and Analysis ...

O tempo consumido por um algoritmo para atingir o objetivo ao qual se propõe, expresso em função do tamanho do problema, é chamado "complexidade tempo" do algoritmo. O comportamento delimitador desta complexidade é chamado de "complexidade tempo assintótica". Conceitos análogos podem ser utilizados para "complexidade memória" e "complexidade memória assintótica".

A complexidade assintótica de um algoritmo é que determina o tamanho máximo do problema que pode ser resolvido por este algoritmo.

Se um algoritmo processa dados de um problema de tamanho n , num tempo cn^2 , para determinada constante $c > 0$, então diz-se que a complexidade tempo deste algoritmo é $O(n^2)$, ou seja, "de ordem n^2 ". Mais precisamente, uma função $g(n)$ é dita ser $O(f(n))$, se existir uma constante c tal que $g(n) \leq c f(n)$ para todo conjunto finito de valores não negativos de n .²

É possível pensar-se que, devido ao tremendo aumento da velocidade de cálculo da geração atual de computadores, a importância de algoritmos eficientes possa ser relegada a segundo plano. Todavia, acontece exatamente o contrário. A complexidade de um algoritmo exerce influência decisiva sobre o tamanho dos problemas a serem resolvidos, influência esta muito mais significativa que aquela proveniente do aumento da velocidade de cálculo dos computadores atuais, como será mostrado a seguir.

² AHO, HOPCROFT & ULLMANN, op. cit ...

Supondo-se disponíveis cinco algoritmos, $A_1 - A_5$, com as seguintes complexidades tempo:

| Algoritmo | Complexidade Tempo |
|-------------|--------------------|
| A_1 | n |
| A_2 | $n \log_2 n$ |
| A_3 | n^2 |
| A_4 | n^3 |
| A_5 | 2^n |

Para o exemplo, a complexidade tempo representará o número de unidades de tempo necessárias para processar uma entrada de tamanho n . Assumindo que uma unidade de tempo seja igual a um mili-segundo, o algoritmo A_1 poderá processar em um segundo uma entrada de tamanho 1000, enquanto o algoritmo A_5 , neste mesmo tempo, não poderá processar uma entrada cujo tamanho seja superior a 9.

O quadro 1 mostra o tamanho dos problemas que poderão ser resolvidos em um segundo, um minuto e uma hora, por cada um destes cinco algoritmos.

QUADRO 1. Limites nos tamanhos dos problemas em função das complexidades tempo dos algoritmos.³

| Algoritmo | Complexidade Tempo | Tamanho máximo do problema | | |
|-----------|-----------------------|----------------------------|-----------------|-------------------|
| | | 1 seg | 1 min | 1 hora |
| A_1 | n | 1000 | 6×10^4 | $3,6 \times 10^6$ |
| A_2 | $n \log n$ | 140 | 4893 | $2,0 \times 10^5$ |
| A_3 | n^2 | 31 | 244 | 1897 |
| A_4 | n^3 | 10 | 39 | 153 |
| A_5 | 2^n | 9 | 15 | 21 |

Supondo-se agora que a próxima geração de computadores venha a ser dez vezes mais rápida que a atual, é apresentado o quadro 2, onde são mostrados os incrementos nos tamanhos dos problemas que poderão ser resolvidos, em decorrência do aumento na velocidade de processamento.

³ AHO, HOPCROFT & ULLMANN, op. cit.

QUADRO 2. Efeito do aumento de dez vezes na velocidade de processamento sobre o tamanho máximo dos problemas.⁴

| Algoritmo | Complexidade tempo | Tamanho máximo do problema | |
|-----------|--------------------|----------------------------|-------------------|
| | | Antes do aumento | Depois do aumento |
| A_1 | n | s_1 | $10s_1$ |
| A_2 | $n \log_2 n$ | s_2 | $\approx 10s_2$ |
| A_3 | n^2 | s_3 | $3,16s_3$ |
| A_4 | n^3 | s_4 | $2,15s_4$ |
| A_5 | 2^n | s_5 | $s_5 + 3,3$ |

Observando o QUADRO 2 notar-se-á que o efeito do aumento de dez vezes na velocidade de processamento, permite apenas um aumento pouco superior a três unidades no tamanho do problema, quando se utiliza o algoritmo A_5 , ao passo em que utilizando-se o algoritmo A_3 , este tamanho pode ser triplicado.

Agora, ao invés do aumento da velocidade computacional, será considerado o efeito da utilização de um algoritmo mais eficiente. Novamente as referências serão feitas ao quadro 1. Usando um minuto como base de análise, a substituição do algoritmo A_4 pelo algoritmo A_3 possibilitará a resolução de problemas 6 ve

⁴ AHO, HOPCROFT & ULLMANN, op. cit.

zes maiores. Substituindo o algoritmo A_4 pelo algoritmo A_2 é possível resolver-se problemas 125 vezes maiores. Estes resultados, são muito mais significativos que aqueles obtidos pelo aumento de dez vezes na velocidade de cálculo. Ao se utilizar uma hora como base de análise, estes resultados tornam-se mais impressionantes ainda. Pode-se então concluir que a complexidade assintótica de um algoritmo é realmente uma importante medida de sua qualidade.

Vale ressaltar no entanto que, para se fazer a seleção de um entre vários algoritmos, é preciso que se tenha em mente o tamanho do problema a ser resolvido, além naturalmente das complexidades dos algoritmos disponíveis. Por exemplo, considere-se que as complexidades tempo dos algoritmos disponíveis sejam na realidade as apresentadas no quadro 3.

QUADRO 3. Complexidade real dos algoritmos.

| Algoritmo | Complexidade Tempo |
|-------------|-----------------------|
| A_1 | $1000\ n$ |
| A_2 | $100\ n\ \log_2 n$ |
| A_3 | $10\ n^2$ |
| A_4 | n^3 |
| A_5 | 2^n |

Desta maneira, o algoritmo A_5 , apesar de, comparado aos

demais apresentar a maior complexidade tempo, deverá ser selecionado quando o objetivo for resolver problemas com tamanho $2 \leq n \leq 9$.

O algoritmo A_2 por sua vez, mostra-se mais adequado para problemas com tamanhos $10 \leq n \leq 58$.

O algoritmo A_2 será superior na faixa de tamanhos $59 \leq n \leq 1024$ e finalmente o algoritmo A_1 que apresenta a menor complexidade tempo, só deverá ser utilizado para resolver problemas cujos tamanhos sejam superiores a 1024.

1.3. Objetivo do Trabalho

Este trabalho visa estudar e analisar o problema do caixeiro viajante, com o objetivo de desenvolver e implementar em computador um algoritmo que apresente baixos níveis de complexidades tempo e memória, e que possa suprir a necessidade atual de algoritmos capazes de resolver problemas de tamanho grande, ($n \geq 100$) obtendo resultados satisfatórios.

C A P Í T U L O I I

2. CONSIDERAÇÕES CONCEITUAIS

Seja $C = \{c_{ij}\}$ uma matriz quadrada que represente a grandeza que se quer analisar, utilizando o modelo do problema do caixeiro viajante, seja ela tempo, distância, custo ou outra.

Qualquer sequência de $p+1$ inteiros, por exemplo, $(1, 2, 3, \dots, n)$, sendo $p \geq n$ e na qual cada um dos n elementos distintos aparece ao menos uma vez e onde o primeiro e o último elemento são idênticos, é chamada circuito.⁵

Um circuito pode ser descrito através de um conjunto ordenado do tipo: $t = (i_1, i_2, i_3, \dots, i_{p-1}, i_p, i_1)$

Uma solução factível para o problema do caixeiro viajante é um circuito.

Uma solução ótima para o mesmo problema é um circuito tal que:

$$z(t) = \sum_{(i,j) \in t'} c_{ij} \text{ é minimizado,}$$

onde $t' = \{(i_1, i_2), (i_2, i_3), \dots, (i_{p-1}, i_p), (i_p, i_1)\}$

é a representação dos pares ordenados de t .

Na terminologia que será adotada, os n elementos serão chamados nós e corresponderão às cidades, os pares ordenados (i, j) serão chamados arcos e ligarão os nós, enquanto a distância do nó i ao nó j será representada por c_{ij} .

Nestes termos, um circuito t é um caminho fechado que passa por cada nó pelo menos uma vez.

⁵

BELLMORE, M. & NEMHAUSER, G.L. The Traveling Salesman ...

O comprimento do circuito, representado por $z(t)$ é a soma dos comprimentos dos arcos a ele pertencentes.

Um sub-circuito s ,

$$s = (i_1, i_2, i_3, \dots, i_k, i_1)$$

é um caminho fechado que não passa por todos os n nós.

2.1. Métodos de Resolução Existentes

Existem várias teorias sobre o problema do caixeiro viajante, baseadas nas quais se fundamentam os diversos algoritmos atualmente existentes.

Os métodos de resolução deste problema podem ser divididos em três partes básicas:

- O ponto de partida
- O esquema de geração de soluções
- A regra de parada

Quando a regra de parada determinar que o término do processo iterativo só se dê quando e somente quando o circuito obtido for ótimo, o método é dito exato.

Quando a regra de parada permitir que o término do processo iterativo possa ocorrer antes da obtenção do circuito ótimo, o método é considerado aproximado.

Nos métodos aproximados, o circuito obtido como solução geralmente depende do ponto de partida, de tal forma que é possível obter-se diversos circuitos diferentes, usando-se diferentes pontos de partida. Pode-se escolher o melhor destes circuitos para solução final do problema.

Como a maioria das regras de parada e das escolhas

dos pontos de partida dependem dos esquemas de geração de soluções, far-se-á uma classificação destes:

2.1.1. Melhoramento Circuito a Circuito

Neste esquema, o ponto de partida é um circuito arbitrário, tal como $t_0 = (1, 2, 3, \dots, n, 1)$.

O procedimento consiste em gerar circuitos cujos comprimentos totais dos arcos sejam inferiores ao comprimento t_0 tal dos arcos do circuito atual, que é tomado como semente. Para a geração é utilizada a redondeza da semente. Por exemplo, t_1 será o circuito gerado a partir de t_0 pelo intercâmbio sucessivo dos elementos $i = 2, 3, \dots, n$ com 1, e que apresente o menor comprimento total dos arcos entre todos os circuitos analisados, inclusive a semente.

Se $t = (p, i_2, i_3, \dots, i_n, p)$, então será t_2 será o melhor circuito (menor comprimento total dos arcos) possível de ser obtido a partir de t_1 através do intercâmbio dos elementos i_j , $j = (2, 3, \dots, n)$ com p .

Uma regra de parada neste caso, pode ser por exemplo, só parar o processo iterativo quando nenhum melhoramento puder ser conseguido.

Pode-se escolher qualquer outro ponto de partida e repetir o procedimento, ou até mesmo aplicar outro procedimento mais sofisticado.

Todos os métodos conhecidos que utilizam este esquema^{6 7 8 9 10} são aproximados.

2.1.2. Construção do Circuito

Neste esquema, o ponto de partida é um nó arbitrário i_1 , a partir do qual vai-se construindo sequências (i_1, i_2, \dots, i_k) por inclusões sucessivas de outros nós na sequência atual, utilizando-se determinados critérios. O procedimento termina quando um circuito for montado. Um critério muito simples para a construção do circuito é o algoritmo do vizinho mais próximo: parte-se de i_1 , em direção ao nó mais próximo i_2 , de onde se segue para o nó mais próximo ainda não visitado, e assim sucessivamente até atingir-se o nó i_n , de onde se retorna para i_1 , completando o circuito.

Obviamente este método é aproximado. No entanto, existem métodos exatos de construção de circuitos, tais como a programação dinâmica^{11 12 13} e o "branch-and-bound".¹⁴

⁶ BARACHET, L.L., Graphic Solution of the Traveling ...

⁷ CROES, G.A., A Method for Solving Traveling ...

⁸ LIN, S., Computer Solution of the Traveling ...

⁹ REITER, S. & SAERMAN G., Discrete Optimizing ...

¹⁰ ROBERTS, S.M & FLORES B., An Engineering Approach ...

¹¹ BELLMAN, R. Dynamic Programming Treatment ...

¹² GONZALES, R.H. Solution to the Traveling ...

¹³ HELD, M. & KARP, R.M. A Dynamic Programming Approach ...

¹⁴ LAWLER, E.L. The Quadratic Assignment Problem ...

2.1.3. Eliminação de Sub-Circuitos

Neste esquema o ponto de partida é a solução ótima do problema de atribuição sobre a matriz C. Se esta solução for um circuito, este circuito será a solução ótima para o problema do caixeiro viajante. Se, no entanto, a solução do problema de atribuição não for um circuito, parte-se para um esquema iterativo de eliminação de sub-circuitos. Existem métodos exatos para eliminação de sub-circuitos, destacando-se entre eles a programação linear inteira^{15 16 17 18 19 20} e o método "branch-and-bound".²¹

¹⁵ BOCK, F. Mathematical Programming Solution ...

¹⁶ DANTZIG, G.B., FULKERSON & JOHNSON, S.M., Solution of a Large ...

¹⁷ DANTZIG, G.B., On a Linear .

¹⁸ MARTINI, G.T. Solving the Traveling Salesman

¹⁹ MILLER, C.E., TUCKER & ZEMLIN, R.A., Integer Programming ...

²⁰ MUDROV, V.I. A Method of Solution of the Traveling ...

²¹ EASTMAN, W.L. Linear Programming ...

C A P Í T U L O I I I

3. MODELO PROPOSTO

O modelo proposto surgiu de uma pesquisa intensa, que se tornou necessária depois do estudo dos algoritmos atuais e da análise do problema do caixeiro viajante.

A pesquisa abrangeu diversos enfoques alternativos, devendo ser ressaltada a teoria dos grafos, que foi o enfoque aparentemente mais promissor, onde se investiu uma grande parte do tempo da pesquisa, e ainda por ter propiciado a descoberta de informações heurísticas que são de importância fundamental para o modelo final.

O algoritmo desenvolvido e implementado sob a teoria dos grafos foi o algoritmo A^* ,²² que garantiu a solução ótima dos problemas analisados, mas que mesmo assim teve que ser abandonado por não atender ao objetivo deste trabalho no tocante ao tamanho dos problemas a serem resolvidos.

Depois desta experiência passou-se a concordar com a literatura, que diz ser o problema do caixeiro viajante um problema que apresenta complexidade tempo exponencial e que os métodos heurísticos não podem ser evitados quando se deseja enfrentá-lo.

Desta maneira as pesquisas se voltaram para os métodos heurísticos, que foram responsáveis por alegrias temporárias e tristezas profundas, todas elas finalmente neutralizadas e compensadas pelo algoritmo a que se chegou.

3.1. O Algoritmo

Projetado para trabalhar com matrizes simétricas o algoritmo é composto de três partes, essencialmente simples, descritas a seguir:

3.1.1. O Ponto de Partida

O algoritmo inicia com um sub-circuito que reflita o conhecimento disponível do problema a ser analisado.

Como excelente heurística para o ponto de partida pode ser utilizado o sub-circuito composto pelos pontos do polígono externo.

3.1.2. O Esquema de Geração de Soluções

Consiste em inserir ao sub-circuito de partida, pontos ainda não visitados, baseado no critério do máximo proveito, descrito a seguir:

1. Sejam i e j dois pontos adjacentes pertencentes ao sub-circuito de partida.

2. Partindo de i , tendo j como objetivo, calcula-se o custo alternativo de ir de i a j através de k , para todo k não pertencente ao sub-circuito, ou seja:

$$\text{MAX_PROV}(i,j,k) = C(i,k) + C(k,j) - C(i,j)$$

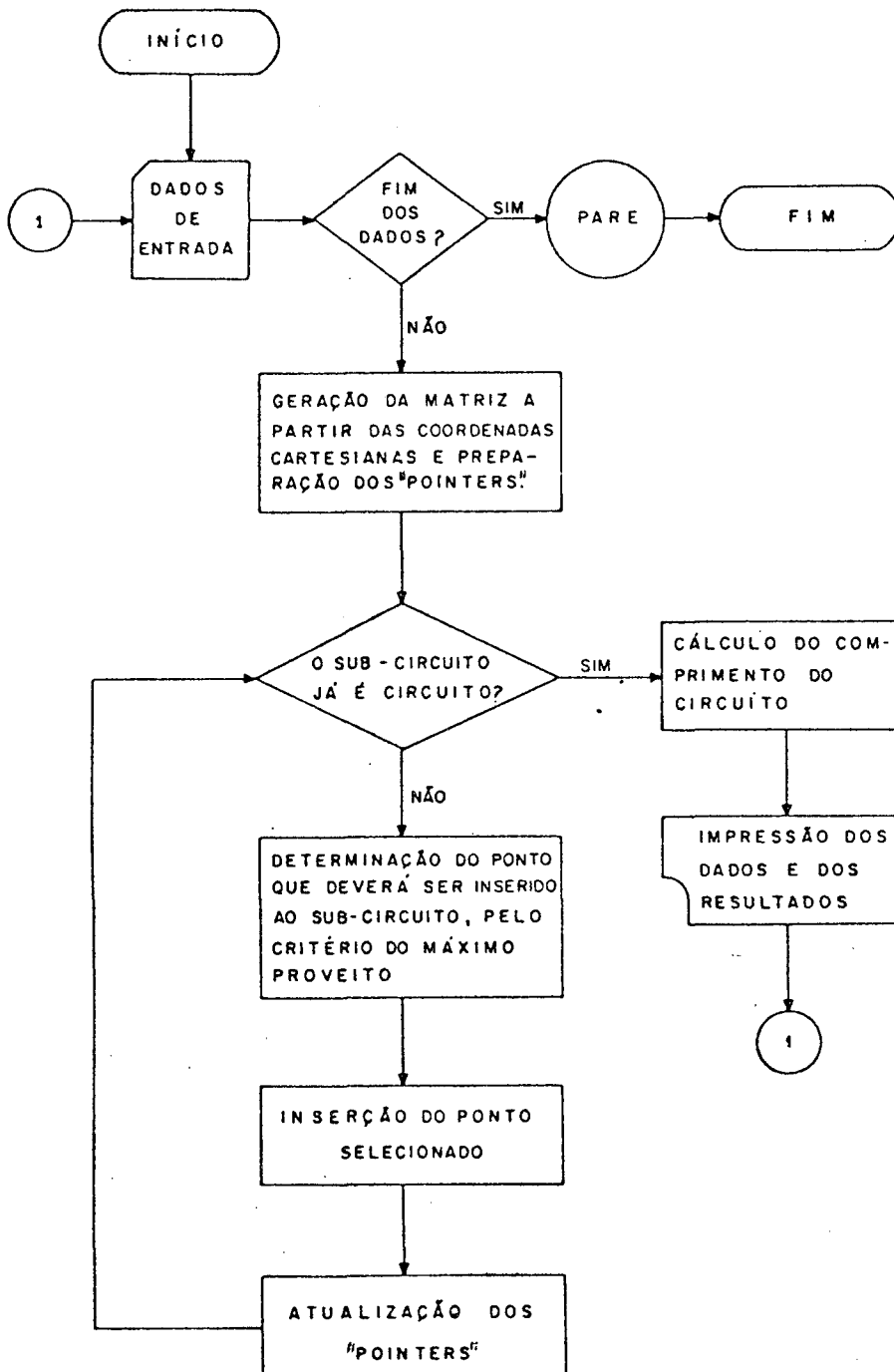
3. Repete-se o passo 2 até que todos os pares de pontos adjacentes do sub-circuito de partida tenham sido considerados, quando então deverá ser selecionado e inserido ao sub-circuito de partida o ponto k associado ao menor $\text{MAX_PROV}(i,j,k)$, pois este propicia o máximo proveito que se pode tirar da situação.

Tendo agora este novo sub-circuito de partida, volta-se ao passo 1.

3.1.3. A Regra de Parada

A regra de parada determina que o esquema de geração de soluções deverá parar quando um circuito for formado.

3.2. O Fluxograma



3.3. Considerações sobre a implementação

Para a implementação deste algoritmo foram utilizados os seguintes recursos:

3.3.1. HARDWARE

- 3.3.1.1. Computador IBM/370-148
- 3.3.1.2. Partição de memória de 256 Kbytes
- 3.3.1.3. Leitora de cartões perfurados
- 3.3.1.4. Impressora com 133 caracteres por linha
- 3.3.1.5. Terminal de vídeo

3.3.2 SOFTWARE

- 3.3.2.1. Sistema operacional OS-VS1, R 6.7
- 3.3.2.2. Compilador PL/I OPTMIZING, V1, R3, PTF 69
- 3.3.2.3. CONVERSATIONAL MONITOR SYSTEM.

3.4. Análise de Resultados

Com o objetivo de calibrar o modelo, foram feitos e xaustivos testes, processando problemas clássicos, problemas teóricos e problemas reais.

Com a finalidade de avaliar o desempenho obtido, a presenta-se o quadro seguinte:

AMOSTRA DOS RESULTADOS OBTIDOS

| PROBLEMAS | NÚMERO DE CIDADES | SOLUÇÃO OBTIDA | SOLUÇÃO ÓTIMA | MELHOR SOLUÇÃO ENCONTRADA | TEMPO DE CPU (SEGUNDOS) |
|-----------|-------------------|----------------|---------------|---------------------------|-------------------------|
| 1 | 5 | 20 | 20 | 20 | 0,89 |
| 2 | 7 | 92 | 92 | 92 | 0,94 |
| 3 | 10 | 200 | 200 | 200 | 1,03 |
| 4 | 16 | 118 | 118 | 118 | 1,13 |
| 5 | 20 | 392 | - | 392 | 1,30 |
| 6 | 36 | 180 | 180 | 180 | 2,36 |
| 7 | 41 | 231 | - | (227) | 3,19 |
| 8 | 50 | 433 | - | 433 | 4,95 |
| 9 | 75 | 552 | - | 552 | 12,49 |
| 10 | 100 | 500 | 500 | 500 | 23,67 |
| 11 | 110 | 550 | 550 | 550 | 31,56 |
| 12 | 200 | 1000 | 1000 | 1000 | 175,51 |

As formulações e respectivas resoluções destes problemas são encontradas no ANEXO 3.

Os mesmos problemas foram processados num computador IBM 4341, instalado na UFSC, utilizando o sistema operacional CMS sob o VM/370, e desta maneira os tempos de CPU gastos nas suas resoluções foram reduzidos em aproximadamente três vezes. O problema das 200 cidades, por exemplo, consumiu exatamente 1 minuto de CPU para a sua resolução.

3.5. Complexidades Tempo e Memória

Através de um modelo de regressão simples e utilizando os dados da amostra apresentada, foi ajustada uma curva para representar a complexidade tempo do algoritmo, sendo obtida a seguinte equação:

$$T = a_0 + a_1 n + a_2 n^2 + a_3 n^3, \text{ onde}$$

$$a_0 = 0.1305012 \quad E + 00$$

$$a_1 = -0.6277299 \quad E - 01$$

$$a_2 = 0.2595460 \quad E - 02$$

$$a_3 = 0.2729445 \quad E - 05$$

T representa o tempo do CPU em segundos e n representa o número de cidades.

Desta maneira, obteve-se um coeficiente de correlação de 0,99983 e uma complexidade tempo $O(n^3)$, devendo ser ressaltado ainda que o coeficiente a_3 , associado ao grau da equação obtida, está muito próximo de zero, e, desta forma, a influência do terceiro grau só começa a ser percebida para valores muito grandes de n.

A complexidade memória por sua vez, é $O(n^2)$, pois o algoritmo aloca apenas a área de memória necessária para armazenar a matriz dos dados e os "pointers".

Vale ressaltar que, para o problema do caixeiro viajante, não é possível obter-se uma complexidade memória de ordem menor que n^2 .

C A P Í T U L O I V

4. CONCLUSÃO

O problema do caixeiro viajante é um destes problemas aos quais estão associados complexidades assintóticas muito altas.²³

Pode-se observar que, ao se utilizar, por exemplo, um método de enumeração explícita, torna-se necessário processar $(n - 1) !/2$ ciclos de operações para achar a solução ótima de um problema simétrico, fato intimamente ligado à complexidade tempo e que inviabiliza este método de resolução, para problemas com tamanhos $n \geq 20$.

A literatura diz que, apesar das décadas de pesquisas, ainda não foi encontrado um só algoritmo com complexidade tempo limitada polinomialmente que garanta a solução ótima do problema do caixeiro viajante.²⁴

Diz ainda que problemas deste tipo podem ser considerados intratáveis,²⁵ e que o enfoque heurístico não pode ser evitado quando se procura resolvê-los.²⁶

O algoritmo desenvolvido neste trabalho apresenta-se como uma ferramenta de valor inestimável para aqueles que necessitam utilizar este modelo para a tomada de decisões.

Não existe internamente ao programa limitação quanto ao tamanho do problema a ser analisado.

O maior problema submetido a processamento, apenas para satisfazer curiosidade, foi de 500 cidades e consumiu mais tempo na formulação que na própria resolução.

²³ AHO, HOPCROFT & ULLMANN, op.cit.

²⁴ AHO, HOPCROFT & ULLMANN, op.cit.

²⁵ AHO, HOPCROFT & ULLMANN, op.cit.

²⁶ LIN, S & KERNIGHAN, B.W. An effective Heuristic ...

BIBLIOGRAFIA UTILIZADA

1. AHO, HOPCROFT & ULLMANN, The Design and Analysis of Computer Algorithms, Addison-Wesley Publishing Company, 1976.
2. BELLMORE, M. & NEMHAUSER, G.L., "The Traveling-Salesman Problem: A Survey". Opns. Res. 16, 538-558 (1968).
3. BARACHET, L.L., "Graphic Solution of the Traveling Salesman Problem," Opns. Res. 5, 841-845 (1957).
4. BEARDWOOD, I., J.H. HALTON, AND J.M. HAMMERSLEY, "The Shortest Path Through Many Points," Proc. Cambridge Phil. - Soc., (Math. and Phys. Sci) 55, 299-327 (1959).
5. BELLMAN, R., "Dynamic Programming Treatment of the Traveling Salesman Problem," J. ACM 9, 61-63 (1962).
6. BOCK, F., "Mathematical Programming Solution of Traveling Salesman Examples," Recent Advances in Mathematical Programming, (R.L. GRAVES AND P. WOLFE, eds) 339-341 McGraw-Hill, New York, 1963.
7. CROES, G.A., "A Method for Solving Traveling Salesman Problems," Opns. Res. 6, 791-812 (1958).
8. DACEY, M.F., "Selection of An Initial Solution for the Traveling Salesman Problem," Opns. Res. 8, 133-134 (1960).
9. DANTZIG, G.B., D.R. FULKERSON, AND S.M. JOHNSON, "Solution of a Large Scale Traveling Salesman Problem," Opns. Res. 2, 393-410 (1954).

10. _____, "On a Linear Programming, Combinatorial Approach to the Traveling Salesman Problem," Opns. Res. 7, 58-66 (1954)
11. DERMAN, C. AND M. KLEIN, "Surveillance of Multicomponent Systems: A Stochastic Traveling Salesman's Problem," Nav. Res. Log. Quart. 13, 103-112 (1966).
12. EASTAMAN, W. L., "Linear Programming with Pattern Constraints," Ph.D., Dissertation, Harvard, 1958.
13. FLEISHMANN, B., "Computational Experience with an Algorithm of Balas", Opns. Res 15, 153-155 (1967).
14. FLOOD, M. M., "The Traveling Salesman Problem", Opns. Res. 4, 61-75 (1956).
15. GILMORE, P. C., AND R. E. GOMORY, "Sequencing a One-State Variable Machine: A Solvable Case of the Traveling Salesman Problem," Opns. Res. 12, 655-679 (1964).
16. GOMORY, R. E., "An Algorithm for Integer Solutions to Linear Programs", Recent Advances in Mathematical Programming (R. L. GRAVES AND P. WOLFE, eds.) 269-302, McGraw-Hill, New York, 1963.
17. GONZALES, R. H., "Solution to the Traveling Salesman Problem by Dynamic Programming on the Hypercube," Tech. Rep. No. 18, O.R. Center, M.I.T., 1962.
18. HARDGRAVE, W. W. AND G. L. NEMHAUSER, "On the Relation Between the Traveling Salesman and the Longest Path Problem", Opns. Res. 10, 647-657 (1962).

19. HILLIER, F.S. & LIEBERMAN, G.L. Operations Research, HoldenDay, Inc. 1974.
20. HELD, M. AND R. M. KARP, "A Dynamic Programming Approach to Sequencing Problems," SIAM 10, 196-210 (1962).
21. KARG, L.L. AND G. L. THOMPSON, "A Heuristic Approach to Solving Traveling Salesman Problems," Management Sci. 10, 225-248 (1964).
22. KNUTH, D. E. Fundamental Algorithms - The Art of Computer Programming, Addison-Wesley Publishing Company, 1973.
23. LIN, S. & KERNIGHAN B.W. An effective heuristic algorithm for the Traveling-Salesman Problem, Bell Telephone Laboratories, Incorporated, Munay Hill, N. J, 1971.
24. _____, AND D. E. WOOD, "Branch-and-Bound Methods: A Survey," Opns. Res. 14, 699-719 (1966).
25. LIN, S., "Computer Solution of the Traveling Salesman Problem," Bell System Tech. J. 44, 2245-2269 (1965).
26. LITTLE, J.D. C. K. G. MURTY, D. W. SWEENEY, AND C. KAREL, "An Algorithm for the Traveling Salesman Problem," Opns. Res. 11, 979-989 (1963).
27. MARTIN, G.T., "An Accelerated Euclidean Algorithm for Integer Linear Programming", Recent Advances in Mathematical Programming (R. L. GRAVES AND P. WOLFE, eds.), 311-318, 1963.
28. _____, "Solving the Traveling Salesman Problem by Integer Linear Programming," CEIR, New York, 1966.

29. MILLER, C. E., A. W. TUCKER, AND R. A. ZEMLIN, "Integer Programming Formulations and Traveling Salesman Problems," J. ACM 7, 326-329.
30. MUDROV, V. I., "A Method of Solution of the Traveling Salesman Problem by Means of Integer Linear Programming (The Problem of Finding the Hamiltonian Paths of Shortest Length in a Complete Graph)" IAOR 5, abstract 3330.1965.
31. NILSSON, N. J. Problems-Solving Methods in Artificial Intelligence, McGraw-Hill Book Company, 1971.
32. ROBERTS, S. M. AND B. FLORES, "An Engineering Approach to the Traveling Salesman Problem," Management Sci. 13, 269-288 (1966).
33. ROTHKOPE, M., "The Traveling Salesman Problem: On the Reduction of Certain Large Problems to Smaller Ones," Opns. Res. 14, 532-533 (1966).
34. SALZ, N. P., "The NORMA: A Possible Basis for a Theory for The Traveling Salesman Problem," Cornell Aero. Lab., Buffalo, N. Y., 1964.
35. _____, "The NORMA: A Theory for the Traveling Salesman Problem," Cornell Aero. Lab., Buffalo, N. Y., 1966.
36. SHAPIRO, D., "Algorithms for the Solution of the Optimal Cost Traveling Salesman Problem," Sc. D. Thesis, Washington University, St. Louis, 1966.
37. STOUDDHAMMER, J. AND M. ASH, "A Sufficiency Solution of the Traveling Salesman Problem," Sys. Aer. Corp., Santa Monica, California, 1966.

A N E X O 1

LISTAGEM DO PROGRAMA

SOURCE LISTING

```
1//===== /
1//
1// THE TRAVELING SALESMAN PROBLEM /
1// ===== /
1//
1// ALGORITHMIC HEURISTIC /
1// OF COMPLEXITY - TIME /
1// LIMITED POLYNOMIALS /
1//
1// LUCIANO R. G. B. RAZ /
1//
1// ===== /
```

```

ITSP1      :PROC OPTIONS(MAIN) RECDPR:

I          DCL MAT_BASE (3,3)          BIN FIXED(15,0) CTL ;
I          DCL MAT_TRAB (3,3)          BIN FIXED(15,0) CTL ;
I          DCL POL_CVXD (3)            BIN FIXED(15,0) CTL ;
I          DCL CCC_CART (3,3)          BIN FIXED(15,0) CTL ;
I          DCL TFSIE                    BIN FIXED(15,0)      ;
I          DCL (PI,PJ)                  BIN FIXED(15,0)      ;
I          DCL CPGAL_ENTRADA            CHAR(6)              ;
I          DCL (GUARDE,P_K_ENTPA)      BIN FIXED(15,0)      ;
I          DCL (FRECUENC,RFF)           BIN FIXED(15,0)      ;
I          DCL (LX1,LY1,LX2,LY2)        BIN FIXED(15,0)      ;
I          DCL BIG_M BIN FIXED(15,0) INIT(32767)            ;
I          DCL BLA_BLA_BLA( )           CHAR(72)            CTL ;
I          CN ENFILE(SYSIN) GO TO FIM;

I      ST:GET LIST(NCLM):

I          ALLOC BLA_BLA_BLA(NCCM):

I          GET SKIP EDIT((BLA_BLA_BLA(1) DO I=1 TO NCCM))(A(72),X(8));
I          PUT PAGE EDIT(BLA_BLA_BLA) (SKIP,X(20),A(72));
I          GET LIST(N,NPP):

I          ALLOC MAT_BASE (N,N+1);
I          ALLOC MAT_TRAB (N,N+1);
I          ALLOC POL_CVXD (N+1);
I          ALLOC CCC_CART(N,3);

I          GET LIST(CPGAL_ENTRADA);

I          IF CPGAL_ENTRADA /= 'COORD' & CPGAL_ENTRADA /= 'MATRIZ'
I              THEN DO;

I              PUT FILE (SYSPRINT) EDIT

```

```

I          (' TSP : A OPCAC DE ENTRADA DE DADOS ('.
I          OPCAC_ENTRADA,') ESTA INCORRETA ')
I          (COL(10),3 A);
I          STOP;
I      END;
I  IF OPCAC_ENTRADA = 'COORDEN'
I      THEN DO:
I          GET LIST(COO_CART);
I          DO L=1 TO N-1;
I              MAT_BASE(L,L)=BIG_N;
I              LX1=COO_CART(L,2);
I              LY1=COO_CART(L,3);
I              DO K=L+1 TO N;
I                  LX2=COO_CART(K,2);
I                  LY2=COO_CART(K,3);
I                  MAT_BASE(L,K),MAT_BASE(K,L)=
I                      SQRT(((LX2-LX1)**2+((LY2-LY1)**2));
I              END;
I          END;
I      END;
I  IF OPCAC_ENTRADA = 'MATRIZ'
I      THEN DO:
I          DO I = 1 TO N;
I              DO J = 1 TO N;
I                  GET LIST (MAT_BASE(I,J));
I                  MAT_BASE(J,I) = MAT_BASE(I,J);
I              END;

```

```

1          END;
1
1          END;
1
1      MAT_BASE(N,N)=BIG_M;
1
1      GET LIST((POL_CVXC(I) DO I=1 TO NPP)) ;
1
1      PUT PAGE EDIT('C A P C S   D E   E N T R A D A')
1
1          (SKIP(2),COL(45),A);
1
1      IF CPGAD_ENTRADA = 'COORDS'
1
1          THEN DO;
1
1              PUT FILE (SYSPRINT) EDIT
1
1                  ('COORDINADAS DAS',N,' CIDADES',(27))'=',' ' '
1
1                  (SKIP(2),COL(47),A,F(4),A,SKIP,COL(47),
1
1                  A,SKIP,A);
1
1              PUT FILE (SYSPRINT) EDIT
1
1                  (CCL_CART)(SKIP,CCL(22),(5)((3)F(5),X(3))) ;
1
1          END;
1
1      IF CPGAD_ENTRADA = 'MATRIZ'
1
1          THEN DO;
1
1              PUT FILE (SYSPRINT) EDIT
1
1                  (' MATRIZ   DAS',N,' CIDADES',(27))'=',' ' '
1
1                  (SKIP(2),COL(47),A,F(4),A,SKIP,COL(47),
1
1                  A,SKIP,A.SKIP);
1
1              DO I = 1 TO N ;
1
1                  DO J = 1 TO N ;
1
1                      PUT FILE (SYSPRINT) EDIT
1
1                          ('A (' ,I,',',',J,') = ',MAT_BASE(I,J))
1
1                          (CCL(22),A,F(3),A,F(3),A,F(5));
1
1              END;
1

```



```

I          END;
I
I      END;
I
I      PUT EDIT('POLIGONO EXTERNO',(16))=' ', ' '
I
I          (SKIP(5),COL(52),A,SKIP,COL(52),A,SKIP,A);
I
I      PUT EDIT((PCL_CVXC(I) DO I=1 TO NPP)) (SKIP,COL(35),(10)F(5));
I
I      MAT_TRAB=MAT_BAST;
I
I      DO I=1 TO NPP-1;
I
I          DO J=1 TO NPP-1;
I
I              MAT_TRAB(PCL_CVXC(I),PCL_CVXC(J))=BIG_M;
I
I          END;
I
I      END;
I
I      ALT:PCCLEND=BIG_M;
I
I      KONT=0;
I
I      DO I=1 TO NPP-1;
I
I          PI=PCL_CVXC(I);
I
I          PJ=PCL_CVXC(I+1);
I
I          LENGTH=1.5*MAT_BAST(PI,PJ);
I
I          DO J=1 TO N;
I
I              T-SIP=MAT_TRAB(PCL_CVXC(I),J);
I
I              IF T-SIP <= LENGTH
I
I                  THEN DO;
I
I                      KONT=KONT+1;
I
I                      P_K_ENTRA=J;
I
I                      RIF=MAT_BAST(PI,P_K_ENTRA)+
I
I                      MAT_BAST(P_K_ENTRA,PJ)-
I
I                      MAT_BAST(PI,PJ);
I
I                      IF RIF < 0 THEN RIF=0;

```

```

I          IF PEF < PEQUENC
I
I              THEN DO:
I
I                  PEQUENC=PEF:
I
I                  MAT_TRAB(PI,N+1)=P_K_ENTRA:
I
I                  GUARDE=I:
I
I              END:
I
I          END:
I
I      END:
I
I      IF KENT > 0 THEN
I
I          DO J=NPP TO GUARDE+1 BY -1:
I
I              PCL_CVXC(J+1)=PCL_CVXC(J):
I
I          END:
I
I      ELSE DO:
I
I          PEQUENC=BIG_M:
I
I          DO I=1 TO NPP-1:
I
I              PI=PCL_CVXC(I):
I
I              PJ=PCL_CVXC(I+1):
I
I              DO J=1 TO N:
I
I                  TESTE=MAT_TRAB(PCL_CVXC(I),J):
I
I                  IF TESTE <= BIG_M
I
I                      THEN DO:
I
I                          P_K_ENTRA=J:
I
I                          REF=MAT_BASE(PI,P_K_ENTRA)+
I
I                          MAT_BASE(P_K_ENTRA,PJ)-
I
I                          MAT_BASE(PI,PJ):
I
I                          IF PEF < 0 THEN REF=0:

```

```

I             IF REF < PEQUENG
I
I             THEN DO:
I
I                 PEQUENG=REF:
I
I                 MAT TRAB (PI,N+1)=P_K_ENTRA:
I
I                 GUARDE=I:
I
I             END:
I
I         END:
I
I     END:
I
I     FINE:
I
I     PCL_CVXC (GUARDE+1)=MAT TRAB (PCL_CVXC (GUARDE),N+1):
I
I     NPP=NPP+1:
I
I     IF NPP=N+1
I
I         THEN DO:
I
I             REF=0:
I
I             DO I=1 TO N:
I
I                 REF=REF+MAT_BASA (PCL_CVXC (I),PCL_CVXC (I+1)):
I
I             END:
I
I             PUT PAGE EDIT ('S G L U C A C', (13)'=', ' ')
I
I                 (SKIP (10),COL (54),A,SKIP,COL (54),A,SKIP,A):
I
I             PUT EDIT (PCL_CVXC) (SKIP,COL (15), (20)F (5)):
I
I             PUT EDIT ('C U S T O =',REF) (SKIP (3),COL (54),A,F (5)):
I
I             GO TO SI:
I
I         END:
I
I     DO J=1 TO NPP-1:
I
I         MAT TRAB (PCL_CVXC (GUARDE+1),PCL_CVXC (J))=BIG_M:
I
I         MAT TRAB (PCL_CVXC (J),PCL_CVXC (GUARDE+1))=BIG_M:

```

I END:

I GL TC ALI:

I FIM: END TSPL:

A N E X O 2

INSTRUÇÕES DE UTILIZAÇÃO

INSTRUÇÕES DE UTILIZAÇÃO

O fornecimento dos dados necessários ao modelo proposto para resolver o problema, pode ser feito de uma maneira extremamente simples e flexível, conforme será apresentado a seguir.

Existem cinco tipos diferentes de dados com os quais trabalha o modelo, que serão relacionados abaixo e descritos em seguida.

Tipos diferentes de dados

1. Número de comentários
2. Comentários
3. Número de nós total e número de nós do polígono externo
4. Opção de fornecimento dos dados dos nós
5. Relação dos nós do polígono externo

Não existe pré-determinação de formatos para os dados de entrada, podendo o usuário fornecê-los livremente conforme lhe pareça mais conveniente.

Descrição dos dados de entrada

1. Número de comentários

Deverá indicar o número de cartões que se deseja utilizar para fazer comentários livres sobre o problema.

2. Comentários

Deverá conter os comentários propriamente ditos, para os quais se utilizarão tantos cartões quantos especificados no item anterior.

3. Número de nós total e número de nós do polígono externo

Deverá conter dois números inteiros, sendo o primeiro deles a quantidade total de nós que simbolizam as cidades, enquanto o segundo indicará a quantidade de nós do polígono externo, a crescido de um.

4. Opção de fornecimento dos dados dos nós

Os dados dos nós podem ser fornecidos de duas maneiras distintas, selecionadas por opção do usuário; uma delas é através da matriz quadrada que representa a grandeza em análise; e a outra através das coordenadas cartesianas dos nós, com as quais o modelo monta a matriz referenciada.

A opção a ser feita é indicada através de uma entre as duas palavras seguintes:

'MATRIZ' ou 'COORDENADAS'.

Se for adotada a opção 'MATRIZ', será preciso fornecer os elementos componentes da mesma, por linha, tornando-se necessário apenas a indicação dos elementos da parte triangular superior, uma vez que existe a simetria. Caso seja adotada a opção 'COORDENADAS', será preciso fornecer os números dos nós acompanhados das suas respectivas coordenadas cartesianas, conforme ilustrado a baixo.

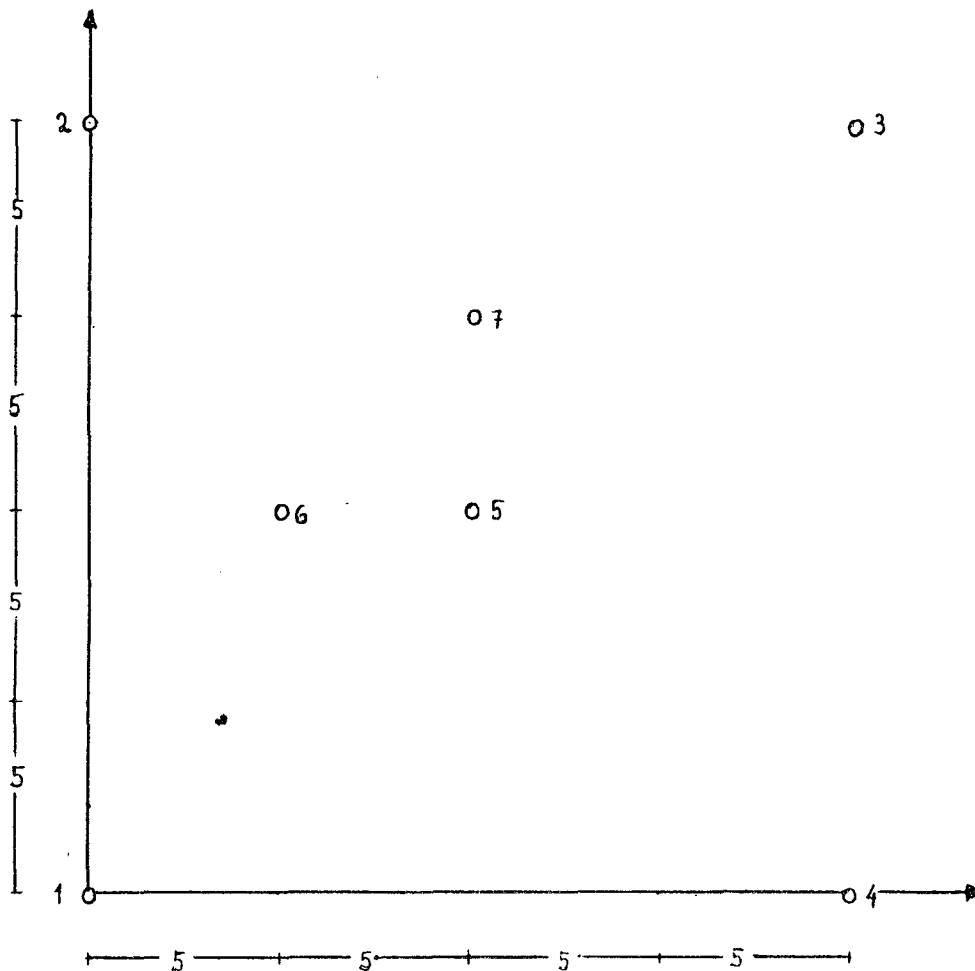
1 X1 Y1 2 X2 Y2 ... ETC.

Os nós deverão ser fornecidos em ordem sequencial crescente.

5. Relação dos nós do polígono externo

Deverá conter os números dos nós que representam o polígono externo.

Com a finalidade de ilustrar tudo o que foi dito, é apresentado um problema e são preparados os dados necessários para a sua resolução através do modelo objeto deste trabalho.



PROBLEMA EXEMPLO.

Dados de Entrada

2

EXEMPLO ILUSTRATIVO DA UTILIZAÇÃO DO MODELO PROPOSTO NO TRABALHO
DA TESE. AUTOR: EUGÊNIO.

7 5

'COORDE'

1 0 0 2 0 20 3 20 20 4 20 0 5 10 10

6 5 10 7 10 15

1 2 3 4 1

/*

- OBS: 1. Num mesmo "job" poderão ser processados tantos problemas
quantos se deseje, bastando para isto repetir a sequência
apresentada anteriormente.
2. Para encerrar o processamento deverá ser colocado um
cartão com /* nas colunas 1 e 2 após o final dos dados.
3. O programa se encontra disponível no CPD da UFSC que
coordenará a sua utilização.

A N E X O 3

PROBLEMAS DE TESTE

D A D O S D E E N T R A D A

COORDENADAS DAS 5 CIDADES

=====

1 0 3 2 2 6 3 4 3 4 7 0 5 8 3

POLIGONO EXTERNO

=====

1 2 5 4 1

S O L U C A O

=====

1 2 3 5 4 1

C U S T O = 20

D A D O S D E E N T R A D A

COORDENADAS DAS 7 CIDADES

=====

1 0 0 2 0 20 3 20 20 4 20 0 5 10 10
6 5 10 7 10 15

POLIGONO EXTERNO

=====

1 2 3 4 1

S O L U C A O

=====

1 5 6 2 7 3 4 1

C U S T O = 92

D A D O S D E E N T R A D A

COORDENADAS DAS 10 CIDADES

=====

| | | | | | | | | | | | | | | |
|---|----|----|---|----|----|---|----|----|---|----|----|----|----|----|
| 1 | 10 | 20 | 2 | 20 | 50 | 3 | 20 | 30 | 4 | 20 | 0 | 5 | 30 | 10 |
| 6 | 40 | 40 | 7 | 40 | 10 | 8 | 50 | 60 | 9 | 50 | 30 | 10 | 70 | 20 |

POLIGONO EXTERNO

=====

1 2 8 10 4 1

S O L U C A O

=====

1 3 2 8 6 9 10 7 5 4 1

C U S T O = 200

D A D O S D E E N T R A D A

COORDENADAS DAS 16 CIDADES

=====

| | | | | | | | | | | | | | | |
|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| 1 | 0 | 10 | 2 | 5 | 15 | 3 | 5 | 5 | 4 | 10 | 20 | 5 | 10 | 0 |
| 6 | 15 | 15 | 7 | 15 | 5 | 8 | 20 | 10 | 9 | 25 | 10 | 10 | 30 | 15 |
| 11 | 30 | 5 | 12 | 35 | 20 | 13 | 35 | 0 | 14 | 40 | 15 | 15 | 40 | 5 |
| 16 | 45 | 10 | | | | | | | | | | | | |

POLIGONO EXTERNO

=====

1 2 4 12 14 16 15 13 5 3

S O L U C A O

=====

1 2 4 6 8 9 10 12 14 16 15 13 11 7 5 3 1

C U S T O = 118

D A D O S D E E N T R A D A

.. COORDENADAS CAS 20 CICADES
=====

| | | | | | | | | | | | | | | |
|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-----|
| 1 | 0 | 0 | 2 | 0 | 40 | 3 | 10 | 70 | 4 | 10 | 20 | 5 | 10 | -10 |
| 6 | 20 | 50 | 7 | 20 | 30 | 8 | 20 | 0 | 9 | 30 | 80 | 10 | 30 | 10 |
| 11 | 40 | 40 | 12 | 40 | 10 | 13 | 50 | 60 | 14 | 50 | 30 | 15 | 50 | -10 |
| 16 | 60 | 40 | 17 | 70 | 60 | 18 | 70 | 20 | 19 | 80 | 40 | 20 | 80 | 10 |

POLIGONO EXTERNO
=====

1 2 3 9 14 17 19 20 15 5
1

S O L U C A O
=====

1 4 7 2 6 3 9 13 11 14 16 17 19 18 20 15 12 10 8 5
1

C U S T O = 392

U A D O S D E E N T R A D A

COORDENADAS DAS 36 CIDADES

=====

| | | | | | | | | | | | | | | |
|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| 1 | 0 | 0 | 2 | 0 | 5 | 3 | 0 | 10 | 4 | 0 | 15 | 5 | 0 | 20 |
| 6 | 0 | 25 | 7 | 5 | 0 | 8 | 5 | 5 | 9 | 5 | 10 | 10 | 5 | 15 |
| 11 | 5 | 20 | 12 | 5 | 25 | 13 | 10 | 0 | 14 | 10 | 5 | 15 | 10 | 10 |
| 16 | 10 | 15 | 17 | 10 | 20 | 18 | 10 | 25 | 19 | 15 | 0 | 20 | 15 | 5 |
| 21 | 15 | 10 | 22 | 15 | 15 | 23 | 15 | 20 | 24 | 15 | 25 | 25 | 20 | 0 |
| 26 | 20 | 5 | 27 | 20 | 10 | 28 | 20 | 15 | 29 | 20 | 20 | 30 | 20 | 25 |
| 31 | 25 | 0 | 32 | 25 | 5 | 33 | 25 | 10 | 34 | 25 | 15 | 35 | 25 | 20 |
| 36 | 25 | 25 | | | | | | | | | | | | |

POLIGONO EXTERNO

=====

| | | | | | | | | | |
|----|----|----|----|----|----|----|----|----|----|
| 2 | 3 | 4 | 5 | 6 | 12 | 18 | 24 | 30 | 36 |
| 35 | 34 | 33 | 32 | 31 | 25 | 19 | 13 | 7 | 1 |
| 2 | | | | | | | | | |

S O L U C A O

=====

| | | | | | | | | | | | | | | | | | | | |
|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|
| 2 | 8 | 14 | 20 | 26 | 27 | 28 | 29 | 23 | 17 | 11 | 10 | 16 | 22 | 21 | 15 | 9 | 3 | 4 | 5 |
| 6 | 12 | 18 | 24 | 30 | 36 | 35 | 34 | 33 | 32 | 31 | 25 | 19 | 13 | 7 | 1 | 2 | | | |

C U S T O = 180

D A O S. D E E N T R A D A

COORDENADAS CAS 41 CIDADES

=====

| | | | | | | | | | | | | | | |
|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| 1 | 0 | 20 | 2 | 5 | 25 | 3 | 5 | 20 | 4 | 5 | 15 | 5 | 10 | 30 |
| 6 | 10 | 25 | 7 | 10 | 20 | 8 | 10 | 15 | 9 | 10 | 10 | 10 | 15 | 35 |
| 11 | 15 | 30 | 12 | 15 | 25 | 13 | 15 | 20 | 14 | 15 | 15 | 15 | 15 | 10 |
| 16 | 15 | 5 | 17 | 20 | 40 | 18 | 20 | 35 | 19 | 20 | 30 | 20 | 20 | 25 |
| 21 | 20 | 20 | 22 | 20 | 15 | 23 | 20 | 10 | 24 | 20 | 5 | 25 | 20 | 0 |
| 26 | 25 | 35 | 27 | 25 | 30 | 28 | 25 | 25 | 29 | 25 | 20 | 30 | 25 | 15 |
| 31 | 25 | 10 | 32 | 25 | 5 | 33 | 30 | 30 | 34 | 30 | 25 | 35 | 30 | 20 |
| 36 | 30 | 15 | 37 | 30 | 10 | 38 | 35 | 25 | 39 | 35 | 20 | 40 | 35 | 15 |
| 41 | 40 | 20 | | | | | | | | | | | | |

POLIGONO EXTERNO

=====

| | | | | | | | | | |
|----|----|----|----|----|----|----|----|----|----|
| 1 | 2 | 5 | 10 | 17 | 26 | 33 | 38 | 41 | 40 |
| 37 | 32 | 25 | 16 | 9 | 4 | 1 | | | |

S O L U C A O

=====

| | | | | | | | | | | | | | | | | | | |
|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| 3 | 7 | 2 | 6 | 13 | 14 | 23 | 30 | 35 | 28 | 29 | 22 | 21 | 20 | 19 | 12 | 5 | 11 | 10 |
| 17 | 26 | 27 | 33 | 34 | 38 | 39 | 41 | 40 | 36 | 37 | 31 | 32 | 24 | 25 | 16 | 15 | 9 | 6 |
| 1 | | | | | | | | | | | | | | | | | | |

C U S T O = 231

U A C C S C E E N T R A C A

COORDENADAS DAS 50 CIDADES

=====

| | | | | | | | | | | | | | | |
|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| 1 | 37 | 52 | 2 | 49 | 49 | 3 | 52 | 64 | 4 | 20 | 20 | 5 | 40 | 30 |
| 6 | 21 | 47 | 7 | 17 | 63 | 8 | 31 | 62 | 9 | 52 | 33 | 10 | 51 | 21 |
| 11 | 42 | 41 | 12 | 51 | 32 | 13 | 5 | 25 | 14 | 12 | 42 | 15 | 30 | 16 |
| 16 | 52 | 41 | 17 | 27 | 23 | 18 | 17 | 33 | 19 | 13 | 13 | 20 | 57 | 58 |
| 21 | 62 | 42 | 22 | 42 | 57 | 23 | 16 | 57 | 24 | 8 | 52 | 25 | 7 | 38 |
| 26 | 27 | 68 | 27 | 30 | 48 | 28 | 43 | 67 | 29 | 58 | 43 | 30 | 58 | 27 |
| 31 | 37 | 69 | 32 | 58 | 46 | 33 | 46 | 10 | 34 | 61 | 33 | 35 | 62 | 63 |
| 36 | 63 | 69 | 37 | 32 | 22 | 38 | 45 | 35 | 39 | 59 | 15 | 40 | 5 | 6 |
| 41 | 10 | 17 | 42 | 21 | 10 | 43 | 5 | 64 | 44 | 30 | 15 | 45 | 39 | 10 |
| 46 | 32 | 39 | 47 | 25 | 32 | 48 | 25 | 55 | 49 | 43 | 23 | 50 | 56 | 37 |

POLIGONO EXTERNO

=====

40 13 43 26 31 36 21 34 39 33
40

S C L U C A C

=====

U 41 13 4 47 12 46 5 38 11 32 1 27 48 6 18 25 14 24 43
S 7 20 8 31 28 22 3 36 35 20 2 29 21 16 50 9 34 30 45
U 39 33 45 15 37 17 44 42 19 40

C U S T O = 432

D A D O S D E E N T R A D A

COORDENADAS DAS 75 CIDADES

=====

| | | | | | | | | | | | | | | |
|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| 1 | 22 | 22 | 2 | 36 | 26 | 3 | 21 | 45 | 4 | 45 | 35 | 5 | 55 | 20 |
| 6 | 33 | 34 | 7 | 50 | 50 | 8 | 55 | 45 | 9 | 23 | 59 | 10 | 40 | 66 |
| 11 | 55 | 65 | 12 | 35 | 51 | 13 | 62 | 35 | 14 | 62 | 57 | 15 | 62 | 24 |
| 16 | 21 | 36 | 17 | 33 | 44 | 18 | 9 | 56 | 19 | 62 | 48 | 20 | 66 | 14 |
| 21 | 44 | 13 | 22 | 26 | 13 | 23 | 11 | 28 | 24 | 7 | 43 | 25 | 17 | 64 |
| 26 | 41 | 46 | 27 | 55 | 34 | 28 | 35 | 16 | 29 | 52 | 25 | 30 | 43 | 26 |
| 31 | 31 | 76 | 32 | 22 | 53 | 33 | 26 | 29 | 34 | 50 | 40 | 35 | 55 | 50 |
| 36 | 54 | 10 | 37 | 60 | 15 | 38 | 47 | 66 | 39 | 20 | 60 | 40 | 30 | 50 |
| 41 | 12 | 17 | 42 | 15 | 14 | 43 | 16 | 19 | 44 | 21 | 43 | 45 | 50 | 30 |
| 46 | 51 | 42 | 47 | 50 | 15 | 48 | 48 | 21 | 49 | 12 | 33 | 50 | 15 | 56 |
| 51 | 29 | 39 | 52 | 54 | 38 | 53 | 55 | 57 | 54 | 67 | 41 | 55 | 10 | 70 |
| 56 | 6 | 25 | 57 | 65 | 27 | 58 | 40 | 60 | 59 | 70 | 64 | 60 | 64 | 4 |
| 61 | 36 | 6 | 62 | 30 | 20 | 63 | 20 | 30 | 64 | 15 | 5 | 65 | 50 | 70 |
| 66 | 57 | 72 | 67 | 45 | 42 | 68 | 38 | 33 | 69 | 50 | 4 | 70 | 66 | 8 |
| 71 | 59 | 5 | 72 | 35 | 60 | 73 | 27 | 24 | 74 | 40 | 20 | 75 | 40 | 37 |

POLIGONO EXTERNO

=====

| | | | | | | | | | |
|----|----|----|----|----|----|----|----|----|----|
| 56 | 24 | 18 | 55 | 31 | 66 | 59 | 54 | 57 | 20 |
| 70 | 60 | 69 | 64 | 56 | | | | | |

S O L U C A O

=====

| | | | | | | | | | | | | | | | | | | |
|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| 23 | 49 | 24 | 18 | 50 | 3 | 44 | 32 | 25 | 55 | 31 | 10 | 72 | 39 | 7 | 58 | 33 | 65 | 11 |
| 59 | 14 | 53 | 7 | 35 | 8 | 19 | 54 | 13 | 57 | 15 | 27 | 52 | 46 | 34 | 67 | 26 | 12 | 40 |
| 51 | 6 | 75 | 68 | 2 | 74 | 30 | 4 | 45 | 29 | 48 | 5 | 37 | 20 | 70 | 60 | 71 | 36 | 69 |
| 21 | 61 | 28 | 62 | 73 | 33 | 16 | 63 | 1 | 22 | 64 | 42 | 43 | 41 | 55 | . | | | |

C U S T O = 552

D A T O S D E E N T R A D A

COORDENADAS DAS 100 CIDADES =====

| | | | | | | | | | | | | | | |
|----|----|----|----|----|----|----|----|----|----|----|----|-----|----|----|
| 1 | 0 | 0 | 2 | 0 | 5 | 3 | 0 | 10 | 4 | 0 | 15 | 5 | 0 | 20 |
| 6 | 0 | 25 | 7 | 0 | 30 | 8 | 0 | 35 | 9 | 0 | 40 | 10 | 0 | 45 |
| 11 | 5 | 0 | 12 | 5 | 5 | 13 | 5 | 10 | 14 | 5 | 15 | 15 | 5 | 20 |
| 16 | 5 | 25 | 17 | 5 | 30 | 18 | 5 | 35 | 19 | 5 | 40 | 20 | 5 | 45 |
| 21 | 10 | 0 | 22 | 10 | 5 | 23 | 10 | 10 | 24 | 10 | 15 | 25 | 10 | 20 |
| 26 | 10 | 25 | 27 | 10 | 30 | 28 | 10 | 35 | 29 | 10 | 40 | 30 | 10 | 45 |
| 31 | 15 | 0 | 32 | 15 | 5 | 33 | 15 | 10 | 34 | 15 | 15 | 35 | 15 | 20 |
| 36 | 15 | 25 | 37 | 15 | 30 | 38 | 15 | 35 | 39 | 15 | 40 | 40 | 15 | 45 |
| 41 | 20 | 0 | 42 | 20 | 5 | 43 | 20 | 10 | 44 | 20 | 15 | 45 | 20 | 20 |
| 46 | 20 | 25 | 47 | 20 | 30 | 48 | 20 | 35 | 49 | 20 | 40 | 50 | 20 | 45 |
| 51 | 25 | 0 | 52 | 25 | 5 | 53 | 25 | 10 | 54 | 25 | 15 | 55 | 25 | 20 |
| 56 | 25 | 25 | 57 | 25 | 30 | 58 | 25 | 35 | 59 | 25 | 40 | 60 | 25 | 45 |
| 61 | 30 | 0 | 62 | 30 | 5 | 63 | 30 | 10 | 64 | 30 | 15 | 65 | 30 | 20 |
| 66 | 30 | 25 | 67 | 30 | 30 | 68 | 30 | 35 | 69 | 30 | 40 | 70 | 30 | 45 |
| 71 | 35 | 0 | 72 | 35 | 5 | 73 | 35 | 10 | 74 | 35 | 15 | 75 | 35 | 20 |
| 76 | 35 | 25 | 77 | 35 | 30 | 78 | 35 | 35 | 79 | 35 | 40 | 80 | 35 | 45 |
| 81 | 40 | 0 | 82 | 40 | 5 | 83 | 40 | 10 | 84 | 40 | 15 | 85 | 40 | 20 |
| 86 | 40 | 25 | 87 | 40 | 30 | 88 | 40 | 35 | 89 | 40 | 40 | 90 | 40 | 45 |
| 91 | 45 | 0 | 92 | 45 | 5 | 93 | 45 | 10 | 94 | 45 | 15 | 95 | 45 | 20 |
| 96 | 45 | 25 | 97 | 45 | 30 | 98 | 45 | 35 | 99 | 45 | 40 | 100 | 45 | 45 |

POLIGONO EXTERNO =====

| | | | | | | | | | |
|----|----|----|----|----|----|----|-----|----|----|
| 2 | 2 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 20 |
| 30 | 40 | 50 | 60 | 70 | 80 | 90 | 100 | 99 | 98 |
| 97 | 96 | 95 | 94 | 93 | 92 | 91 | 81 | 71 | 61 |
| 51 | 41 | 31 | 21 | 11 | 1 | 2 | | | |

S C L U C A C =====

| | | | | | | | | | | | | | | | | | | | |
|---|-----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| 2 | 12 | 22 | 32 | 42 | 52 | 62 | 72 | 82 | 92 | 84 | 85 | 86 | 87 | 88 | 89 | 79 | 69 | 59 | 49 |
| 9 | 29 | 19 | 18 | 17 | 16 | 15 | 14 | 24 | 34 | 44 | 54 | 64 | 65 | 66 | 67 | 57 | 47 | 37 | 36 |
| 6 | 56 | 55 | 45 | 35 | 25 | 26 | 27 | 28 | 38 | 48 | 58 | 68 | 78 | 77 | 76 | 75 | 74 | 73 | 63 |
| 3 | 43 | 23 | 22 | 12 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 20 | 30 | 40 | 50 | 60 | 70 | 80 |
| 0 | 100 | 99 | 98 | 97 | 96 | 95 | 94 | 93 | 92 | 91 | 81 | 71 | 61 | 51 | 41 | 31 | 21 | 11 | 1 |
| 2 | | | | | | | | | | | | | | | | | | | |

C L S T C = 500

D A D O S C E E N T R A D A

COORDENADAS DAS 110 CIDADES

| | | | | | | | | | | | | | | |
|-----|----|----|-----|----|----|-----|----|----|-----|----|----|-----|----|----|
| 1 | 0 | 0 | 2 | 0 | 5 | 3 | 0 | 10 | 4 | 0 | 15 | 5 | 0 | 20 |
| 6 | 0 | 25 | 7 | 0 | 30 | 8 | 0 | 35 | 9 | 0 | 40 | 10 | 0 | 45 |
| 11 | 5 | 0 | 12 | 5 | 5 | 13 | 5 | 10 | 14 | 5 | 15 | 15 | 5 | 20 |
| 16 | 5 | 25 | 17 | 5 | 30 | 18 | 5 | 35 | 19 | 5 | 40 | 20 | 5 | 45 |
| 21 | 10 | 0 | 22 | 10 | 5 | 23 | 10 | 10 | 24 | 10 | 15 | 25 | 10 | 20 |
| 26 | 10 | 25 | 27 | 10 | 30 | 28 | 10 | 35 | 29 | 10 | 40 | 30 | 10 | 45 |
| 31 | 15 | 0 | 32 | 15 | 5 | 33 | 15 | 10 | 34 | 15 | 15 | 35 | 15 | 20 |
| 36 | 15 | 25 | 37 | 15 | 30 | 38 | 15 | 35 | 39 | 15 | 40 | 40 | 15 | 45 |
| 41 | 20 | 0 | 42 | 20 | 5 | 43 | 20 | 10 | 44 | 20 | 15 | 45 | 20 | 20 |
| 46 | 20 | 25 | 47 | 20 | 30 | 48 | 20 | 35 | 49 | 20 | 40 | 50 | 20 | 45 |
| 51 | 25 | 0 | 52 | 25 | 5 | 53 | 25 | 10 | 54 | 25 | 15 | 55 | 25 | 20 |
| 56 | 25 | 25 | 57 | 25 | 30 | 58 | 25 | 35 | 59 | 25 | 40 | 60 | 25 | 45 |
| 61 | 30 | 0 | 62 | 30 | 5 | 63 | 30 | 10 | 64 | 30 | 15 | 65 | 30 | 20 |
| 66 | 30 | 25 | 67 | 30 | 30 | 68 | 30 | 35 | 69 | 30 | 40 | 70 | 30 | 45 |
| 71 | 35 | 0 | 72 | 35 | 5 | 73 | 35 | 10 | 74 | 35 | 15 | 75 | 35 | 20 |
| 76 | 35 | 25 | 77 | 35 | 30 | 78 | 35 | 35 | 79 | 35 | 40 | 80 | 35 | 45 |
| 81 | 40 | 0 | 82 | 40 | 5 | 83 | 40 | 10 | 84 | 40 | 15 | 85 | 40 | 20 |
| 86 | 40 | 25 | 87 | 40 | 30 | 88 | 40 | 35 | 89 | 40 | 40 | 90 | 40 | 45 |
| 91 | 45 | 0 | 92 | 45 | 5 | 93 | 45 | 10 | 94 | 45 | 15 | 95 | 45 | 20 |
| 96 | 45 | 25 | 97 | 45 | 30 | 98 | 45 | 35 | 99 | 45 | 40 | 100 | 45 | 45 |
| 101 | 0 | 50 | 102 | 5 | 50 | 103 | 10 | 50 | 104 | 15 | 50 | 105 | 20 | 50 |
| 106 | 25 | 50 | 107 | 30 | 50 | 108 | 35 | 50 | 109 | 40 | 50 | 110 | 45 | 50 |

POLIGONO EXTERNO

| | | | | | | | | | |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 101 |
| 102 | 103 | 104 | 105 | 106 | 107 | 108 | 109 | 110 | 100 |
| 55 | 58 | 97 | 56 | 55 | 54 | 53 | 92 | 91 | 81 |
| 71 | 61 | 51 | 41 | 31 | 21 | 11 | 1 | 2 | |

S C L U C A C

| | | | | | | | | | | | | | | | | | | | |
|----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|----|----|----|----|----|----|----|----|
| 2 | 12 | 22 | 32 | 42 | 52 | 62 | 72 | 82 | 83 | 84 | 85 | 86 | 87 | 88 | 89 | 90 | 80 | 70 | 60 |
| 50 | 40 | 30 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 24 | 34 | 44 | 54 | 64 | 65 | 66 | 67 | 68 | 58 |
| 48 | 38 | 37 | 36 | 46 | 47 | 57 | 56 | 55 | 45 | 35 | 25 | 26 | 27 | 28 | 29 | 39 | 49 | 59 | 69 |
| 79 | 78 | 77 | 76 | 75 | 74 | 73 | 63 | 53 | 43 | 33 | 23 | 13 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| 10 | 101 | 102 | 103 | 104 | 105 | 106 | 107 | 108 | 109 | 110 | 100 | 99 | 98 | 97 | 96 | 95 | 94 | 93 | 92 |
| 91 | 81 | 71 | 61 | 51 | 41 | 31 | 21 | 11 | 1 | 2 | | | | | | | | | |

C U S T O = 550

D A D C S D E E N T R A C A

COORDENADAS DAS 200 CIDADES

=====

| | | | | | | | | | | | | | | |
|-----|----|----|-----|----|----|-----|----|----|-----|----|----|-----|----|----|
| 1 | 0 | 0 | 2 | 0 | 5 | 3 | 0 | 10 | 4 | 0 | 15 | 5 | 0 | 20 |
| 6 | 0 | 25 | 7 | 0 | 30 | 8 | 0 | 35 | 9 | 0 | 40 | 10 | 0 | 45 |
| 11 | 5 | C | 12 | 5 | 5 | 13 | 5 | 10 | 14 | 5 | 15 | 15 | 5 | 20 |
| 16 | 5 | 25 | 17 | 5 | 30 | 18 | 5 | 35 | 19 | 5 | 40 | 20 | 5 | 45 |
| 21 | 10 | C | 22 | 10 | 5 | 23 | 10 | 10 | 24 | 10 | 15 | 25 | 10 | 20 |
| 26 | 10 | 25 | 27 | 10 | 30 | 28 | 10 | 35 | 29 | 10 | 40 | 30 | 10 | 45 |
| 31 | 15 | C | 32 | 15 | 5 | 33 | 15 | 10 | 34 | 15 | 15 | 35 | 15 | 20 |
| 36 | 15 | 25 | 37 | 15 | 30 | 38 | 15 | 35 | 39 | 15 | 40 | 40 | 15 | 45 |
| 41 | 20 | C | 42 | 20 | 5 | 43 | 20 | 10 | 44 | 20 | 15 | 45 | 20 | 20 |
| 46 | 20 | 25 | 47 | 20 | 30 | 48 | 20 | 35 | 49 | 20 | 40 | 50 | 20 | 45 |
| 51 | 25 | C | 52 | 25 | 5 | 53 | 25 | 10 | 54 | 25 | 15 | 55 | 25 | 20 |
| 56 | 25 | 25 | 57 | 25 | 30 | 58 | 25 | 35 | 59 | 25 | 40 | 60 | 25 | 45 |
| 61 | 30 | C | 62 | 30 | 5 | 63 | 30 | 10 | 64 | 30 | 15 | 65 | 30 | 20 |
| 66 | 30 | 25 | 67 | 30 | 30 | 68 | 30 | 35 | 69 | 30 | 40 | 70 | 30 | 45 |
| 71 | 35 | C | 72 | 35 | 5 | 73 | 35 | 10 | 74 | 35 | 15 | 75 | 35 | 20 |
| 76 | 35 | 25 | 77 | 35 | 30 | 78 | 35 | 35 | 79 | 35 | 40 | 80 | 35 | 45 |
| 81 | 40 | C | 82 | 40 | 5 | 83 | 40 | 10 | 84 | 40 | 15 | 85 | 40 | 20 |
| 86 | 40 | 25 | 87 | 40 | 30 | 88 | 40 | 35 | 89 | 40 | 40 | 90 | 40 | 45 |
| 91 | 45 | C | 92 | 45 | 5 | 93 | 45 | 10 | 94 | 45 | 15 | 95 | 45 | 20 |
| 96 | 45 | 25 | 97 | 45 | 30 | 98 | 45 | 35 | 99 | 45 | 40 | 100 | 45 | 45 |
| 101 | 50 | 0 | 102 | 50 | 5 | 103 | 50 | 10 | 104 | 50 | 15 | 105 | 50 | 20 |
| 106 | 50 | 25 | 107 | 50 | 30 | 108 | 50 | 35 | 109 | 50 | 40 | 110 | 50 | 45 |
| 111 | 55 | C | 112 | 55 | 5 | 113 | 55 | 10 | 114 | 55 | 15 | 115 | 55 | 20 |
| 116 | 55 | 25 | 117 | 55 | 30 | 118 | 55 | 35 | 119 | 55 | 40 | 120 | 55 | 45 |
| 121 | 60 | 0 | 122 | 60 | 5 | 123 | 60 | 10 | 124 | 60 | 15 | 125 | 60 | 20 |
| 126 | 60 | 25 | 127 | 60 | 30 | 128 | 60 | 35 | 129 | 60 | 40 | 130 | 60 | 45 |
| 131 | 65 | C | 132 | 65 | 5 | 133 | 65 | 10 | 134 | 65 | 15 | 135 | 65 | 20 |
| 136 | 65 | 25 | 137 | 65 | 30 | 138 | 65 | 35 | 139 | 65 | 40 | 140 | 65 | 45 |
| 141 | 70 | 0 | 142 | 70 | 5 | 143 | 70 | 10 | 144 | 70 | 15 | 145 | 70 | 20 |
| 146 | 70 | 25 | 147 | 70 | 30 | 148 | 70 | 35 | 149 | 70 | 40 | 150 | 70 | 45 |
| 151 | 75 | C | 152 | 75 | 5 | 153 | 75 | 10 | 154 | 75 | 15 | 155 | 75 | 20 |
| 156 | 75 | 25 | 157 | 75 | 30 | 158 | 75 | 35 | 159 | 75 | 40 | 160 | 75 | 45 |
| 161 | 80 | C | 162 | 80 | 5 | 163 | 80 | 10 | 164 | 80 | 15 | 165 | 80 | 20 |
| 166 | 80 | 25 | 167 | 80 | 30 | 168 | 80 | 35 | 169 | 80 | 40 | 170 | 80 | 45 |
| 171 | 85 | C | 172 | 85 | 5 | 173 | 85 | 10 | 174 | 85 | 15 | 175 | 85 | 20 |
| 176 | 85 | 25 | 177 | 85 | 30 | 178 | 85 | 35 | 179 | 85 | 40 | 180 | 85 | 45 |
| 181 | 90 | C | 182 | 90 | 5 | 183 | 90 | 10 | 184 | 90 | 15 | 185 | 90 | 20 |
| 186 | 90 | 25 | 187 | 90 | 30 | 188 | 90 | 35 | 189 | 90 | 40 | 190 | 90 | 45 |
| 191 | 95 | C | 192 | 95 | 5 | 193 | 95 | 10 | 194 | 95 | 15 | 195 | 95 | 20 |
| 196 | 95 | 25 | 197 | 95 | 30 | 198 | 95 | 35 | 199 | 95 | 40 | 200 | 95 | 45 |

POLIGONO EXTERNO

=====

| | | | | | | | | | |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 20 |
| 30 | 40 | 50 | 60 | 70 | 80 | 90 | 100 | 110 | 120 |
| 130 | 140 | 150 | 160 | 170 | 180 | 190 | 200 | 199 | 198 |
| 197 | 196 | 195 | 194 | 193 | 192 | 191 | 181 | 171 | 161 |
| 151 | 141 | 131 | 121 | 111 | 101 | 91 | 81 | 71 | 61 |
| 51 | 41 | 31 | 21 | 11 | 1 | 2 | | | |

S C L U C A O
=====

| | | | | | | | | | | | | | | | | | | | |
|----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| 2 | 12 | 22 | 32 | 42 | 52 | 62 | 72 | 82 | 92 | 102 | 112 | 122 | 132 | 142 | 152 | 162 | 172 | 182 | 183 |
| 4 | 185 | 186 | 187 | 188 | 189 | 179 | 169 | 159 | 149 | 139 | 129 | 119 | 109 | 99 | 89 | 79 | 69 | 59 | 49 |
| 9 | 29 | 19 | 18 | 17 | 16 | 15 | 14 | 24 | 34 | 44 | 54 | 64 | 74 | 84 | 94 | 104 | 114 | 124 | 134 |
| 4 | 154 | 164 | 165 | 166 | 167 | 157 | 147 | 137 | 127 | 117 | 107 | 97 | 87 | 77 | 67 | 57 | 47 | 37 | 36 |
| 6 | 56 | 66 | 76 | 86 | 96 | 106 | 116 | 126 | 136 | 146 | 156 | 155 | 145 | 135 | 125 | 115 | 105 | 95 | 85 |
| 5 | 65 | 55 | 45 | 35 | 25 | 26 | 27 | 28 | 38 | 48 | 58 | 68 | 78 | 88 | 98 | 108 | 118 | 128 | 138 |
| 8 | 158 | 168 | 178 | 177 | 176 | 175 | 174 | 173 | 163 | 153 | 143 | 133 | 123 | 113 | 103 | 93 | 83 | 73 | 63 |
| 3 | 43 | 33 | 23 | 13 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 20 | 30 | 40 | 50 | 60 | 70 | 80 |
| 10 | 100 | 110 | 120 | 130 | 140 | 150 | 160 | 170 | 180 | 190 | 200 | 199 | 189 | 179 | 169 | 159 | 149 | 139 | 129 |
| 1 | 181 | 171 | 161 | 151 | 141 | 131 | 121 | 111 | 101 | 91 | 81 | 71 | 61 | 51 | 41 | 31 | 21 | 11 | 1 |
| 2 | | | | | | | | | | | | | | | | | | | |

C U S T O = 1000